Nomor 1

Soal:

Selesaikan sistem persamaan linear (SPL) berikut menggunakan metode iterasi Jacobi dan Gauss-Seidel dengan toleransi $\varepsilon = 10^{-3}$ dalam norm ℓ_∞.

$$\begin{aligned}
4x_1 - x_2 - x_4 &= 0,\\
-x_1 + 4x_2 - x_3 - x_5 &= 5,\\
- x_2 + 4x_3 - x_6 &= 0,\\
-x_1 + 4x_4 - x_5 &= 6,\\
- x_2 - x_4 + 4x_5 - x_6 &= -2,\\
- x_3 - x_5 + 4x_6 &= 6.
\end{aligned}$$

Teori:

ITERASI JACOBI

Skema iterasi Jacobi:

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}\, x_j^{(k-1)}\right), \quad i = 1, 2, \ldots, n; \quad k = 1, 2, \ldots$$

Algoritma Iterasi Jacobi

Deskripsi

Untuk menyelesaikan sistem persamaan linear (SPL) $A\boldsymbol{x} = \boldsymbol{b}$, dengan A adalah matriks koefisien $n \times n$, $\boldsymbol{b}$ vektor konstanta $n \times 1$, dan $\boldsymbol{x}$ vektor $n \times 1$ yang akan dicari.

Input

A, $\boldsymbol{b}$, hampiran awal $\boldsymbol{Xo} = \boldsymbol{x}^{(0)}$, toleransi T, dan maksimum iterasi N

Output

Hampiran solusi $x_1, x_2, \ldots, x_n$ atau pesan '*maksimum iterasi terlampaui*'

Langkah-langkah

1. Set $k = 1$
2. $While\ k \leq N$ do (langkah 3 – 6)
   3. $For\ i = 1, \ldots, n$

      Set $x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij} Xo_j\right)$
   4. Jika $\|\boldsymbol{x} - \boldsymbol{Xo}\| < T$, maka

      Output $(x_1, x_2, \ldots, x_n)$

      $STOP$
   5. Set $k = k + 1$
   6. $For\ i = 1, \ldots, n$

      Set $Xo_i = x_i$

7. If (k > N), maka
   Output (‘$Maksimum\ iterasi\ terlampaui$’) $STOP$

ITERASI GAUSS-SEIDEL

Skema iterasi Gauss-Seidel:

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}\, x_j^{(k-1)}\right), \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots$$

Algoritma Iterasi Gauss-Seidel

Deskripsi

Untuk menyelesaikan sistem persamaan linear (SPL) $A\boldsymbol{x} = \boldsymbol{b}$, dengan A adalah matriks koefisien $n \times n$, $\boldsymbol{b}$ vektor konstanta $n \times 1$, dan $\boldsymbol{x}$ vektor $n \times 1$ yang akan dicari.

Input

$A, \boldsymbol{b}$, hampiran awal $\boldsymbol{Xo} = \boldsymbol{x}^{(0)}$, toleransi T dan maksimum iterasi N

Output

Hampiran solusi $x_1, x_2, \dots, x_n$ atau pesan ‘*maksimum iterasi terlampaui*’

Langkah-langkah

1. Set $k = 1$
2. $While$ $k \le N$ do (langkah 3 – 6)
   3. For $i = 1, \dots, n$
      Set $x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}\, x_j - \sum_{j=i+1}^{n} a_{ij}\, Xo_j\right)$
   4. If $\|\boldsymbol{x} - \boldsymbol{Xo}\| < T$
      Output $(x_1, x_2, \dots, x_n)$
      $STOP$
   5. Set $k = k + 1$
   6. For $i = 1, \dots, n$
      Set $Xo_i = x_i$
7. If (k > N), maka
   Output (‘$Maksimum\ iterasi\ terlampaui$’) $STOP$

Implementasi Program

Perhitungan galat dapat menggunakan galat absolut seperti dalam algoritma di atas maupun galat relatif:

$$\frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|}{\|\boldsymbol{x}^{(k)}\|}$$

Dalam implementasi program, perhitungan galat yang digunakan yaitu galat relatif dalam norm ℓ_∞.

Fungsi Jacobi

```
function [X,g,H] = Jacobi(A,b,Xo,T,N)
%---------------------------------------------------------
% Iterasi Jacobi untuk menyelesaikan SPL Ax=b
% Contoh Pemanggilan
%   [X,g,H] = jacobi(A,b,Xo,T,N)
% Input
%   A   : matriks koefisien
%   b   : vektor konstanta (ruas kanan)
%   Xo  : vektor solusi awal
%   T   : batas toleransi
%   N   : maksimum iterasi
% Output
%   X   : vektor solusi
%   g   : vektor galat hampiran
%   H   : matriks dg baris vektor2 hampiran selama iterasi
%----------------------------------------------------------
H = Xo';
n = length(b);
X = Xo;
for k = 1 : N
    for i = 1 : n
        X(i) = b(i) - A(i,[1:i-1,i+1:n])*Xo([1:i-1,i+1:n])) /A(i,i);
    end
    g = abs(X-Xo);
    err = norm(g,inf);
    relerr = err/norm(X,inf);
    Xo = X;
    H = [H; Xo'];
    if relerr < T
        break
    end
end
```

Pemanggilan Fungsi Jacobi

```
A = [4 -1 0 -1 0 0; -1 4 -1 0 -1 0; 0 -1 4 0 0 -1;
     -1 0 0 4 -1 0; 0 -1 0 -1 4 -1; 0 0 -1 0 -1 4];
b = [0; 5; 0; 6; -2; 6];
Xo = [0; 0; 0; 0; 0; 0];
T = 1e-3;
N = 25;
[X,g,H] = Jacobi(A,b,Xo,T,N);
disp('Solusi Hampiran SPL Metode Iterasi Jacobi:')
disp(X')
disp('Solusi Hampiran Selama Iterasi :')
disp('      x1         x2         x3         x4         x5         x6')
disp('    ----------------------------------------------------------')
disp(H)
```

Output Program

```
Solusi Hampiran SPL Iterasi Jacobi:
    0.9991    1.9994    0.9991    1.9996    0.9988    1.9996

Solusi Hampiran Selama Iterasi :
```

x1	x2	x3	x4	x5	x6
0	0	0	0	0	0
0	1.2500	0	1.5000	-0.5000	1.5000
0.6875	1.1250	0.6875	1.3750	0.5625	1.3750
0.6250	1.7344	0.6250	1.8125	0.4688	1.8125
0.8867	1.6797	0.8867	1.7734	0.8398	1.7734
0.8633	1.9033	0.8633	1.9316	0.8066	1.9316
0.9587	1.8833	0.9587	1.9175	0.9417	1.9175
0.9502	1.9648	0.9502	1.9751	0.9296	1.9751
0.9850	1.9575	0.9850	1.9699	0.9787	1.9699
0.9819	1.9872	0.9819	1.9909	0.9743	1.9909
0.9945	1.9845	0.9945	1.9890	0.9923	1.9890
0.9934	1.9953	0.9934	1.9967	0.9907	1.9967
0.9980	1.9944	0.9980	1.9960	0.9972	1.9960
0.9976	1.9983	0.9976	1.9988	0.9966	1.9988
0.9993	1.9979	0.9993	1.9985	0.9990	1.9985
0.9991	1.9994	0.9991	1.9996	0.9988	1.9996

Fungsi Gauss-Seidel

```
function [X,g,H] = GaussSeidel(A,b,Xo,T,N)
%--------------------------------------------------------
% Iterasi Gauss-Seidel untuk menyelesaikan SPL Ax=b
% Contoh Pemanggilan
%   [X,g,H] = GaussSeidel(A,b,Xo,T,N)
% Input
%   A   : matriks koefisien
%   b   : vektor konstanta (ruas kanan)
%   Xo  : vektor solusi awal
%   T   : batas toleransi
%   N   : maksimum iterasi
% Output
%   X   : vektor solusi
%   g   : vektor galat hampiran
%   H   : matriks dg baris vektor2 hampiran selama iterasi
%---------------------------------------------------------
H = Xo';
n = length(b);
X = Xo;
for k = 1 : N
    for i = 1 : n
        X(i) = (b(i)-A(i,[1:i-1])*X(1:i-1)-A(i,[i+1:n])*Xo(i+1:n))/A(i,i);
    end
    g = abs(X-Xo);
    err = norm(g,inf);
    relerr = err/norm(X,inf);
    Xo = X;
    H = [H; Xo'];
    if relerr < T
        break
    end
end
```

Pemanggilan Fungsi Jacobi

```
A = [4 -1 0 -1 0 0; -1 4 -1 0 -1 0; 0 -1 4 0 0 -1;
     -1 0 0 4 -1 0; 0 -1 0 -1 4 -1; 0 0 -1 0 -1 4];
b = [0; 5; 0; 6; -2; 6];
Xo = [0; 0; 0; 0; 0; 0];
T = 1e-3;
N = 25;
[X,g,H] = GaussSeidel(A,b,Xo,T,N);
disp('Solusi Hampiran SPL Metode Iterasi Gauss-Seidel:')
disp(X')
disp('Solusi Hampiran Selama Iterasi :')
disp('      x1         x2         x3         x4         x5         x6')
disp('    ---------------------------------------------------------')
disp(H)
```

Output Program

```
Solusi Hampiran SPL Metode Iterasi Gauss-Seidel :
    0.9995    1.9996    0.9998    1.9997    0.9998    1.9999

Solusi Hampiran Selama Iterasi :
      x1         x2         x3         x4         x5         x6
    ---------------------------------------------------------
         0         0         0         0         0         0
         0    1.2500    0.3125    1.5000    0.1875    1.6250
    0.6875    1.5469    0.7930    1.7188    0.7227    1.8789
    0.8164    1.8330    0.9280    1.8848    0.8992    1.9568
    0.9294    1.9391    0.9740    1.9572    0.9633    1.9843
    0.9741    1.9778    0.9905    1.9843    0.9866    1.9943
    0.9905    1.9919    0.9966    1.9943    0.9951    1.9979
    0.9966    1.9971    0.9987    1.9979    0.9982    1.9992
    0.9987    1.9989    0.9995    1.9992    0.9994    1.9997
    0.9995    1.9996    0.9998    1.9997    0.9998    1.9999
```

Analisa Hasil

Solusi eksak SPL di atas yaitu $(x_1, x_2, x_3, x_4, x_5, x_6) = (1,\ 2,\ 1,\ 2,\ 1,\ 2)$. Dari hasil output program dua metode iterasi di atas menunjukkan bahwa:

1. Hasil iterasi Gauss-Seidel lebih cepat konvergen dibanding hasil iterasi Jacobi, dapat dilihat dari jumlah iterasinya. Iterasi Jacobi membutuhkan 15 iterasi untuk mencapai toleransi minimum, sedangkan iterasi Gauss-Seidel cukup 9 iterasi saja.
2. Jika dibandingkan dengan solusi eksak, solusi hampiran iterasi Gauss-Seidel lebih akurat dibanding hasil iterasi Jacobi.

Kesimpulan

Metode iterasi Gauss-Seidel telah berhasil memperkecil error dan mempercepat kekonvergenan solusi dibanding iterasi Jacobi, karena metode ini memang didesain untuk memperbaiki metode iterasi Jacobi.

Nomor 2

Soal:

Terdapat sistem persamaan linear (SPL):

$$\begin{aligned} 2x_1 - x_2 + x_3 &= -1, \\ 2x_1 + 2x_2 + 2x_3 &= 4, \\ -x_1 - x_2 + 2x_3 &= -5 \end{aligned}$$

yang mempunyai solusi (1, 2, -1).

a. Tunjukkan bahwa $\rho(T_j) = \frac{\sqrt{5}}{2} > 1$,

b. Tunjukkan bahwa metode iterasi Jacobi dengan hampiran awal $\boldsymbol{x}^{(\mathbf{0})} = \mathbf{0}$ gagal menghasilkan hampiran yang baik setelah 25 iterasi,

c. Tunjukkan bahwa $\rho(T_g) = \frac{1}{2}$,

d. Gunakan metode iterasi Gauss-Seidel dengan hampiran awal $\boldsymbol{x}^{(\mathbf{0})} = \mathbf{0}$ untuk menghampiri solusi SPL dengan toleransi 10^{-5} dalam norm ℓ_∞.

Jawab:

Diketahui SPL $A\boldsymbol{x} = \boldsymbol{b}$,

$$\begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 4 \\ -5 \end{pmatrix}$$

a. Akan ditunjukkan bahwa $\rho(T_j) = \frac{\sqrt{5}}{2} > 1$

$T_j = D^{-1}(L+U)$, dimana

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \ L = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \ U = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}, \text{ diperoleh } D^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

$$\text{Maka } T_j = D^{-1}(L+U) = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} \left(\begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} \right) = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

$$\det(T_j - \lambda I) = \begin{vmatrix} -\lambda & \frac{1}{2} & -\frac{1}{2} \\ -1 & -\lambda & -1 \\ \frac{1}{2} & \frac{1}{2} & -\lambda \end{vmatrix} = -\left(\lambda^3 + \frac{5}{4}\lambda\right) = -\lambda\left(\lambda^2 + \frac{5}{4}\right) = 0$$

Diperoleh:

$$\lambda_1 = 0, \qquad \lambda_2 = \frac{\sqrt{5}}{2} i, \qquad \lambda_3 = -\frac{\sqrt{5}}{2} i$$

Maka $\rho(T_j) = maks\left\{|0|, \left|\frac{\sqrt{5}}{2}\right|, \left|-\frac{\sqrt{5}}{2}\right|\right\} = \frac{\sqrt{5}}{2} > 1$ (terbukti).

b. Dengan menggunakan fungsi Jacobi pada program nomor 1 dengan toleransi 10^{-5} dalam norm ℓ_∞ untuk menyelesaikan SPL di atas diperoleh output program:

Output Program

```
Solusi Hampiran SPL Iterasi Jacobi:
  -20.8279    2.0000  -22.8279

Solusi Hampiran Selama Iterasi :
      x1         x2         x3
    --------------------------
         0          0          0
   -0.5000    2.0000   -2.5000
    1.7500    5.0000   -1.7500
    2.8750    2.0000    0.8750
    0.0625   -1.7500   -0.0625
   -1.3438    2.0000   -3.3438
    2.1719    6.6875   -2.1719
    3.9297    2.0000    1.9297
   -0.4648   -3.8594    0.4648
   -2.6621    2.0000   -4.6621
    2.8311    9.3242   -2.8311
    5.5776    2.0000    3.5776
   -1.2888   -7.1553    1.2888
   -4.7220    2.0000   -6.7220
    3.8610   13.4441   -3.8610
    8.1526    2.0000    6.1526
   -2.5763  -12.3051    2.5763
   -7.9407    2.0000   -9.9407
    5.4703   19.8814   -5.4703
   12.1759    2.0000   10.1759
   -4.5879  -20.3517    4.5879
  -12.9698    2.0000  -14.9698
    7.9849   29.9397   -7.9849
   18.4623    2.0000   16.4623
   -7.7311  -32.9246    7.7311
  -20.8279    2.0000  -22.8279
```

Hasil di atas menunjukkan bahwa iterasi Jacobi tidak berhasil menemukan solusi hampiran yang baik karena sangat jauh dari solusi eksaknya.

c. Akan ditunjukkan bahwa $\rho(T_g) = \frac{1}{2}$

$T_g = (D-L)^{-1}U$, dimana

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad L = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$$

$$D - L = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix}, \text{ maka } (D-L)^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

Sehingga diperoleh $T_g = (D-L)^{-1}U = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$

$$\det(T_g - \lambda \mathrm{I}) = \begin{vmatrix} -\lambda & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2}-\lambda & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2}-\lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\left(\frac{1}{2}+\lambda\right) & -\frac{1}{2} \\ 0 & -\left(\frac{1}{2}+\lambda\right) \end{vmatrix} = -\lambda\left(\frac{1}{2}+\lambda\right)^2 = 0$$

Diperoleh:

$\lambda_1 = 0, \qquad \lambda_2 = -\dfrac{1}{2}, \qquad \lambda_3 = -\dfrac{1}{2}$

Maka $\rho\left(T_g\right) = maks\ \left\{|0|,\ \left|-\frac{1}{2}\right|,\ \left|-\frac{1}{2}\right|\right\} = \frac{1}{2}$ (terbukti)

d. Dengan menggunakan fungsi Gauss-Seidel pada program nomor 1 dengan toleransi 10^{-5} dalam norm ℓ_∞ untuk menyelesaikan SPL di atas diperoleh output program:

Output Program

```
Solusi Hampiran SPL Metode Iterasi Gauss-Seidel :
    1.0000    2.0000   -1.0000

Solusi Hampiran Selama Iterasi :
      x1        x2        x3
    --------------------------
         0         0         0
   -0.5000    2.5000   -1.5000
    1.5000    2.0000   -0.7500
    0.8750    1.8750   -1.1250
    1.0000    2.1250   -0.9375
    1.0313    1.9063   -1.0313
    0.9688    2.0625   -0.9844
    1.0234    1.9609   -1.0078
    0.9844    2.0234   -0.9961
    1.0098    1.9863   -1.0020
    0.9941    2.0078   -0.9990
    1.0034    1.9956   -1.0005
    0.9980    2.0024   -0.9998
    1.0011    1.9987   -1.0001
    0.9994    2.0007   -0.9999
    1.0003    1.9996   -1.0000
    0.9998    2.0002   -1.0000
    1.0001    1.9999   -1.0000
    0.9999    2.0001   -1.0000
    1.0000    2.0000   -1.0000
```

Hasil di atas menunjukkan bahwa iterasi Gauss-Seidel berhasil menemukan solusi hampiran yang baik pada iterasi ke-23.

Nomor 3

Soal:

Cari akar-akar dari sistem persamaan non-linear berikut menggunakan metode Newton:

$$f_1(x_1, x_2) = 0.5\sin(x_1x_2) - 0.25\frac{x_2}{\pi} - 0.5x_1$$

$$f_2(x_1, x_2) = \left(1 - \frac{0.25}{\pi}\right)(\exp(2x_1) - e) + e\frac{x_2}{\pi} - 2ex_1$$

Jawab:

Dari sistem non-linear di atas dapat dibentuk matriks Jacobi dengan cara:

$$J(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f_1}{x_1} & \frac{\partial f_1}{x_2} \\ \frac{\partial f_2}{x_1} & \frac{\partial f_2}{x_2} \end{pmatrix}$$

Dimana,

$$\frac{\partial f_1}{x_1} = 0.5 \cdot x_2 \cdot \cos(x_1x_2) - 0.5$$

$$\frac{\partial f_1}{x_2} = 0.5 \cdot x_1 \cdot \cos(x_1x_2) - \frac{0.25}{\pi}$$

$$\frac{\partial f_2}{x_1} = \left(1 - \frac{0.25}{\pi}\right) \cdot 2 \cdot \exp(2x_1) - 2e$$

$$\frac{\partial f_2}{x_2} = \frac{e}{\pi}$$

Algoritma Metode Newton

Deskripsi

Untuk menghampiri solusi sistem persamaan non-linear $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$, dengan memberikan sebuah hampiran awal $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$.

Input

hampiran awal ($\boldsymbol{x}$), toleransi T, dan maksimum iterasi N

Output

Hampiran solusi $x_1, x_2, \dots, x_n$ atau pesan '*maksimum iterasi terlampaui*'

Langkah-langkah

1. Set $k = 1$
2. $While\ k \leq N$ do (langkah 3 – 7)
   3. Hitung $\boldsymbol{F}(\boldsymbol{x})$ dan $J(\boldsymbol{x})$
   4. Selesaikan sistem persamaan linear $J(\boldsymbol{x})\boldsymbol{y} = -\boldsymbol{F}(\boldsymbol{x})$
   5. Set $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{y}$
   6. Jika $||\boldsymbol{y}|| < T$, maka

      Output $(x_1, x_2, \dots, x_n)$

      $STOP$

7. Set $k = k + 1$
8. If (k > N), maka

   Output (‘*Maksimum iterasi terlampaui*’) *STOP*

Implementasi Program

Fungsi Newton

```
function [X,H] = Newton(X,T,N)
%---------------------------------------------------------
% Metode Newton untuk menyelesaikan sistem pers. non-linear
% Input
%    X    : vektor tebakan awal
%    T    : batas toleransi
%    N    : maksimum iterasi
% Output
%    X    : vektor solusi
%    H    : matriks hampiran solusi selama iterasi
%----------------------------------------------------------
k = 1;
H = X';
while k <= N
    [y1,dx11,dx21] = f1(X);
    [y2,dx12,dx22] = f2(X);
    F = [y1; y2];
    J = [dx11 dx21; dx12 dx22];
    y = inv(J)*-F;
    X = X + y;
    H = [H; X'];
    if norm(y) < T
        break
    end
    k = k + 1;
end
```

Fungsi 1

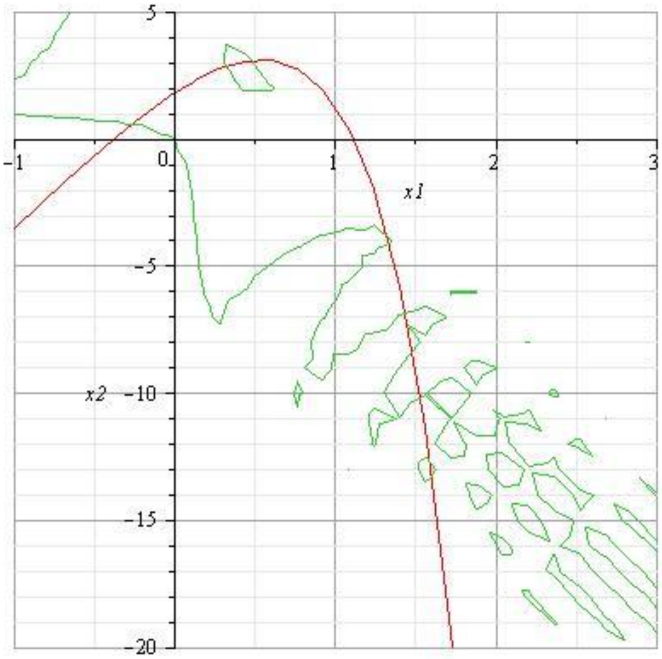
```
function [y1,dx1,dx2] = f1(x)
%---------------------------------------------------------
% f1 = 0.5sin(x1x2)-0.25(x2/pi)-0.5x1
% Input
%    x    : titik
% Output
%    y1   : nilai fungsi f1 di x
%    dx1  : nilai turunan f1 terhadap x1 di x
%    dx2  : nilai turunan f1 terhadap x2 di x
%----------------------------------------------------------
    y1 = 0.5*sin(x(1)*x(2)) - 0.25*x(2)/pi - 0.5*x(1);
    dx1 = 0.5*x(2)*cos(x(1)*x(2)) - 0.5;
    dx2 = 0.5*x(1)*cos(x(1)*x(2)) - 0.25*pi;
```

Fungsi 2

```
function [y1,dx1,dx2] = f2(x)
%---------------------------------------------------------
% f2 = (1-(0.25/pi))(exp(2x1)-e)+e*x2/pi-2*e*x1
```

```
% Input
%   x   : titik
% Output
%   y1  : nilai fungsi f2 di x
%   dx1 : nilai turunan f2 terhadap x1 di x
%   dx2 : nilai turunan f2 terhadap x2 di x
%----------------------------------------------------------
    y1 = (1-(0.25/pi))*(exp(2*x(1))-exp(1))+exp(1)*x(2)/pi-2*exp(1)*x(1);
    dx1 = (1-(0.25/pi))*2*exp(2*x(1)) - 2*exp(1);
    dx2 = exp(1)/pi;
```

Berikut ini adalah gambar dua persamaan non-linear di atas menggunakan Maple dalam interval $-1 \le x_1 \le 3$ dan $-20 \le x_2 \le 5$.



Dari gambar di atas dapat kita lihat terdapat beberapa titik potong antara dua kurva dan itulah akar-akarnya. Selanjutnya akan dilakukan pencarian akar-akar dengan dengan mencoba beberapa tebakan awal seperti disajikan dalam tabel berikut:

No	Tebakan Awal		Akar-akar	
	x1	x2	x1	x2
1	1	-3	1.2943	-3.1360
2	1	-4	1.2943	-3.1363
3	1	-5	1.4339	-6.8201
4	1.5	-7	1.4340	-6.8212
5	1.5	-10	1.5305	-10.2019
6	2	-11	1.6046	-13.3624
7	-0.5	1	-0.2601	0.6251
8	0.5	3	0.5002	3.1417
9	1.4	-12	1.5305	-10.2016

Hasil percobaan menunjukkan ditemukan beberapa akar-akar yaitu:

(1.2943, -3.1363), (1.4340, -6.8212), (1.5305, -10.2019), (1.6046, -13.3624), (-0.2601, 0.6251), (0.5002, 3.1417).